

Remallado de Terrenos Multirresolución

M. Cifuentes^{1,2}, A. Corbellin², M. Rolón², G. Rodríguez², P. Lotito², A. Clausse^{2,3}
cifuyente@exa.unicen.edu.ar

¹Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

²Universidad Nacional del Centro de la Provincia de Buenos Aires

³Comisión Nacional de Energía Atómica

Abstract

This paper proposes a new methodology for the multiresolution visualization of earth's topography defined by many adjacent real topographical models. The strategy refines progressively the model subdividing it into rectangular triangles, building a binary hierarchy generated by the progressive refining of the model, using the error indicator. It combines the observer location, the local curvature of the zone and the height fields and the triangle size put before. The triangulation applies refining and triangle grouping processes, adding or deleting detail to the representation.

We also apply previously developed ideas to real elevation digital models.

Key words: Multiresolution, topographical model, local curvature, triangle strip, real elevation, DEM (Digital Elevation Model).

Resumen

Se propone una nueva metodología para la visualización multirresolución de topografía terrestre definida por varios modelos topográficos reales colindantes. La estrategia refina progresivamente al modelo subdividiéndolo en triángulos rectángulos construyendo una jerarquía binaria generada por el refinamiento progresivo del modelo, según un indicador de error que combina ubicación del observador, curvatura local de la zona y del campo de alturas y tamaño deseable en los triángulos. La triangulación aplica procesos de refinamiento y agrupación de triángulos, agregando o sacando detalle a la representación. Además, se aplican ideas desarrolladas anteriormente a modelos digitales de elevación reales.

Palabras Claves: Multiresolución, modelo topográfico, curvatura local, triangle strip, elevación real, MED (Modelo de Elevación Digital).

1 INTRODUCCIÓN

En los sistemas de información geográfica, la porción de superficie terrestre a representar proviene de grandes modelos topográficos digitales, razón por la cual es necesario incorporar técnicas que faciliten su ágil manipulación y/o visualización en tiempo real. La literatura actual propone técnicas de generación de modelos multiresolución que posibilitan la extracción del mismo terreno con resolución variada. Por ejemplo, representar con mucho detalle aquellas regiones del terreno que se encuentran cercanas al ojo del observador y modelar con unos pocos polígonos las regiones alejadas [1-8].

Este trabajo propone una estrategia que agiliza la visualización como también la transmisión en tiempo real de extensos terrenos. La idea es construir una aproximación multiresolución de la superficie basada en la ubicación del observador, inclinación de la dirección de vista, e indicador de la curvatura local de la región. Los triángulos que la componen se organizan en un *árbol binario* del que se extrae la triangulación final de la superficie: una malla poligonal multiresolución conforme.

Además, el algoritmo propuesto para la generación de la malla almacena los polígonos en un *triangle strip* como una serie de triángulos conectados que comparten vértices. Esta estructura acelera la carga en memoria, la visualización y disminuye el espacio necesario en disco. Las placas gráficas disponibles la optimizan, constituyendo el modo más eficiente de describir un objeto en *computers graphics*. La compactación comienza después de introducidos los tres primeros vértices del primer triángulo en el strip. Por cada nuevo triángulo se incorpora un nuevo vértice, los dos vértices restantes se comparten con el del último triángulo introducido.

Finalmente, se aplicó para la visualización de modelos poligonales correspondientes a modelos digitales de elevación reales (MDE) y sintéticos, un algoritmo de reconocimiento de vértices colineales el cual minimiza la redundancia de los mismos entre triángulos adyacentes.

2 METODOLOGÍA PARA LA SIMPLIFICACIÓN DE TERRENOS

En la actualidad es una realidad el modelado computacional de topografía terrestre obtenida a partir de modelos digitales de elevación (MDE), grillas regulares compuestas por millones de cotas que describen una porción de la superficie terrestre. Generalmente, para modelar terrenos extensos se agrupan MDEs que definen sectores colindantes de la superficie en estudio. En principio, surge la necesidad de automatizar la fusión de MDEs con la implicancia del tratamiento interactivo de grillas voluminosas con tamaños y/o resoluciones disímiles [9]. En consecuencia, la región geográfica de interés surge como resultado de la aplicación de un *algoritmo de encaje* especialmente implementado y cuya funcionalidad, por ejemplo, facilita la unión de varios MDEs representantes de zonas adyacentes del terreno en estudio. La magnitud de la información de los MDEs constituye un problema al momento de visualizar la topografía; no obstante, es posible reducir este volumen procesando únicamente la cápsula convexa de cotas inmersas en la vista. De todas maneras, es conveniente construir modelos simplificados que empleen un reducido conjunto de cotas que conserven la apariencia en la representación.

Básicamente, la idea es reducir la cantidad de polígonos que definen originalmente al modelo construyendo una aproximación multiresolución. El algoritmo de simplificación refina incrementalmente el terreno aplicando criterios basados en la curvatura local del campo escalar de alturas definido por el MDE, la posición del observador y el tamaño del triángulo. El grado de detalle se determina imponiendo una cota superior a la acumulación de un indicador extensivo (i.e. proporcional a la superficie) de curvatura corregido con la distancia al observador y el tamaño aceptable de los triángulos. El criterio es que las regiones con más curvatura y más cercanas al

observador requieren mayor nivel de detalle. De esta forma los lugares con baja curvatura pueden ser simplificados sin perder calidad [4].

El terreno o campo de alturas se transforma en una matriz cuadrada de dimensión $2^k \times 2^k$, donde k es el máximo nivel de subdivisión permitido. El proceso comienza con los dos triángulos iniciales representantes del terreno y calcula para cada triángulo la curvatura local de sus cotas. Tales triángulos corresponden al hijo izquierdo y al derecho de un árbol binario que se usará para el almacenamiento de la triangulación. El indicador de error basado en la curvatura local establece hasta cuándo es necesario continuar con el refinamiento de cada región. Finalmente, los nodos terminales o frontera del árbol conforman la malla poligonal que se está visualizando en un dado instante (ver figura 1).

Otra problemática es la conformidad de la malla poligonal a nivel intrazonal e interzonal; es decir, evitar la presencia de roturas en la trama de la malla provocados por triángulos adyacentes con nivel de subdivisión disímiles o la articulación zonas adyacentes que provienen de distintos MDE. Para lograr una triangulación conforme, la jerarquía binaria no necesita restringirse por niveles como es el caso de los *quadtree* [7], simplemente recurre al tratamiento de cadenas de dependencias entre vértices. Permite generar alternativas de triangulación más flexibles, en las que se puede refinar localmente un triángulo sin afectar a los triángulos adyacentes. Esta problemática se soluciona incorporando una cota ficticia en la mitad del lado más largo del triángulo, su magnitud se calcula interpolando las cotas u extremos del lado. Las cotas ficticias se transforman en cotas reales como se indicará en la sección 3.

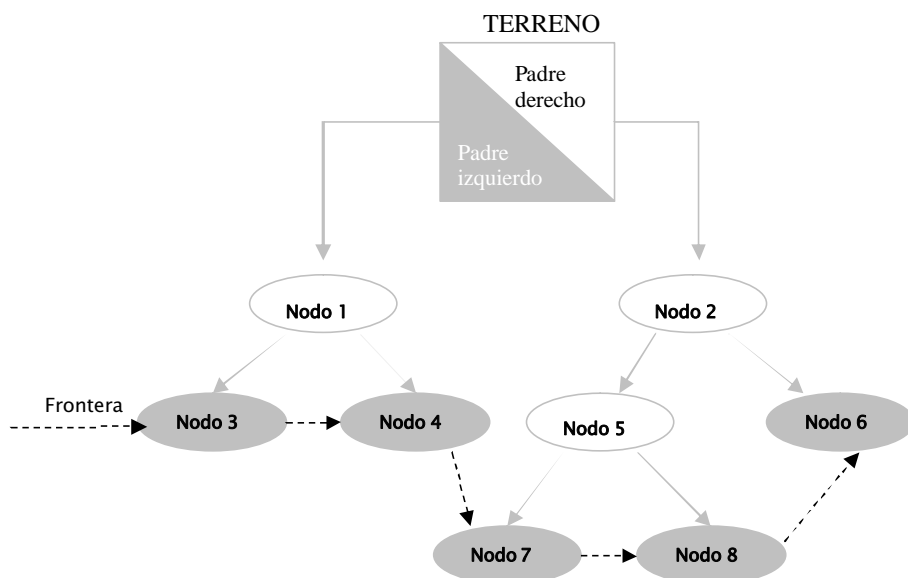


Figura 1- Representación jerárquica del terreno.

El algoritmo de simplificación comprueba en $O(n \log n)$ si el indicador de error sugerido se verifica en los nodos terminales (o frontera). Si el nodo cumple con las condiciones impuestas en el indicador de error, inmediatamente se procede a insertar el triángulo en una frontera auxiliar que reemplazará a la actual, en caso contrario el proceso subdivide al triángulo en dos triángulos (sus descendientes), incrementa en uno la cantidad de triángulos a representar, los inserta en la frontera auxiliar y repite el mismo proceso con cada uno de ellos.

Por otra parte, al desplazarse el observador, sucede que las zonas antes cercanas ya no necesitan el detalle que conservan y pueden remallarse [11]. Este proceso recursivo se inicia en los nodos terminales (frontera) y continúa con los padres, ascendiendo en el árbol hasta alcanzar el nivel de detalle deseado.

La selección automática de detalle se logra mediante la conjugación de tres factores relevantes que otorgan la información necesaria desde tres ópticas fundamentales: homogeneidad de zonas (factor de curvatura), tamaño válido del triángulo (factor tamaño) y posición relativa del observador con respecto a la zona (factor distancia). Con respecto a la homogeneidad, se evalúa la curvatura local de cada triángulo y se compara con un umbral derivado del cálculo según la ubicación del observador. Este criterio compara la desviación de la geometría intrínseca de la superficie respecto a la geometría del plano. Con respecto al tamaño del triángulo, conviene establecerlo como una medida proporcionada con la longitud de la diagonal del rectángulo que representa al terreno en análisis. Por último, el factor distancia resultará una medida relativa al valor de la distancia máxima.

En las siguientes abstracciones procedurales se puede visualizar cómo funciona la selección automática de detalle, donde *triangular* es una función recursiva que incrementa el nivel de detalle en una zona.

```
agregar_detalle (vértice * observador) {
    para todas las zonas en la frontera
        si cercanía_al_observador(zona, observador) and
           tamaño_triángulo_válido( zona ) and
           no planaridad( zona, observador) entonces
            triangular(zona_hija_izquierda, observador)
            triangular(zona_hija_derecha, observador)
        sino
            actualizar_frontera (zona);
}
```

```
sacar_detalle (matriz * zona, vértice * observador) {
    si cercanía_al_observador(zona, observador) or
       tamaño_triángulo_válido( zona ) or
       no planaridad( zona, observador) entonces
        restaurar_strip(zona);
        sacar_detalle(zona, observador);
}
```

En todos los algoritmos planteados se hace referencia al tamaño del triángulo, lo que nos da una referencia que indica cuán lejos estamos del tamaño límite impuesto. Finalmente, la distancia del observador a la zona debe ser aceptable para proporcionar una cuota de realidad dada por el alcance de una vista ficticia.

3 IMPLEMENTACIÓN

Los vértices de los triángulos se organizan en un *triangle strip* debido a que son construcciones eficientemente soportadas por el hardware gráfico y ocupan menos espacio que otras representaciones. Los triángulos surgen de tomar grupos de tres vértices contiguos del *triangle strip*, formando tres tipos de triángulos A, B y C (ver figura 2). Este agrupamiento genera triángulos línea, que hemos eliminado verificando la colinealidad de los vértices previamente.

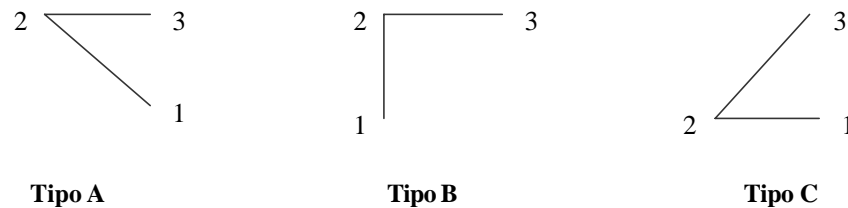


Figura 2- Tipos de triángulos A, B y C. Subdivisión de un triángulo de tipo B.

La subdivisión de un triángulo (o secuencia de vértices) de tipo A genera dos nuevos triángulos o secuencias de tipos B y C respectivamente. Similarmente, cuando el triángulo es de tipo B, se generan dos triángulos de tipo A y C. La figura 3 muestra un triángulo de tipo B cuya secuencia generadora es $\{1, 2, 3\}$, en un paso intermedio se agrega a la secuencia el segundo vértice quedando $\{1, 2, 3, 2, 3\}$; por último, se genera la secuencia generadora de los dos triángulos hijos $\{1, 2, 4, 2, 3\}$. Cada secuencia determina el strip de un triángulo o una sucesión de triángulos. Finalmente, si el triángulo es de tipo C, se subdivide en dos subsecuencias de tipos A y B [1].

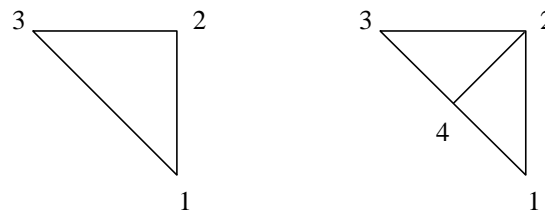


Figura 3 – Subdivisión de un triángulo de tipo B.

Los nodos del árbol conservan el número de triángulo por el cual se ordena la estructura que representa a la frontera, el tipo de triángulo (tipo A, B o C), y los punteros al *triangle strip*. También se conservan las coordenadas del vértice, su altura y un *estado* valuado en *seleccionado* (S) cuando el vértice posee la cota real, *ficticio* (F) cuando la cota surge de la interpolación de los extremos del segmento y *no seleccionado* (D) para desmarcar vértices, útil al proceso de compresión. Los algoritmos de compresión y descompresión restauran el *triangle strip*, decrementando o incrementando la longitud de la secuencia de vértices debido a que la cantidad de triángulos disminuye o se agrega.

Cuando se está por subdividir el triángulo, es muy importante fijarse qué va a pasar con la zona enfrentada en pos de evitar cracks y aproximar con realismo el terreno. Inicialmente, el *estado* del vértice 4 presente en la figura 4 es *ficticio*, por lo tanto su altura proviene de la interpolación. No obstante, como es compartido por triángulos adyacentes que han sufrido diferentes niveles de subdivisión, tendrá asociado un vértice dependiente: el vértice 6. Cuando su vecino adyacente

también se subdivide, el *estado* del vértice 4 será el del *seleccionado*, su valor tendrá el de la cota real y no tendrá dependiente, ya que el crack ha sido salvado en ambas zonas vecinas. De esta forma, mediante lógica de posibles estados del vértice, se evita rápidamente la formación de cracks, impidiendo el manejo de cadenas de dependencias recursivas entre zonas vecinas.

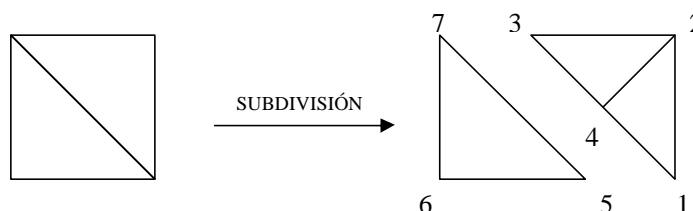


Figura 4 – Dependencias de vértices para eliminación de cracks

4 ANÁLISIS DE RESULTADOS

El algoritmo propuesto se aplicó a la región geográfica demarcada en la Figura 5, precisamente la franja terrestre definida entre los 67° y los 71° Oeste correspondiente a los 35° Sur. Tal información fue obtenida a partir de cinco archivos HGT disponibles en www.goldensoftware.com que representan a una región compuesta por 14,4 millones de triángulos. En este formato, cada archivo cubre áreas de 1° x 1° con resolución adecuada, definidos por 1201x1201 celdas o 2,88 millones de triángulos.

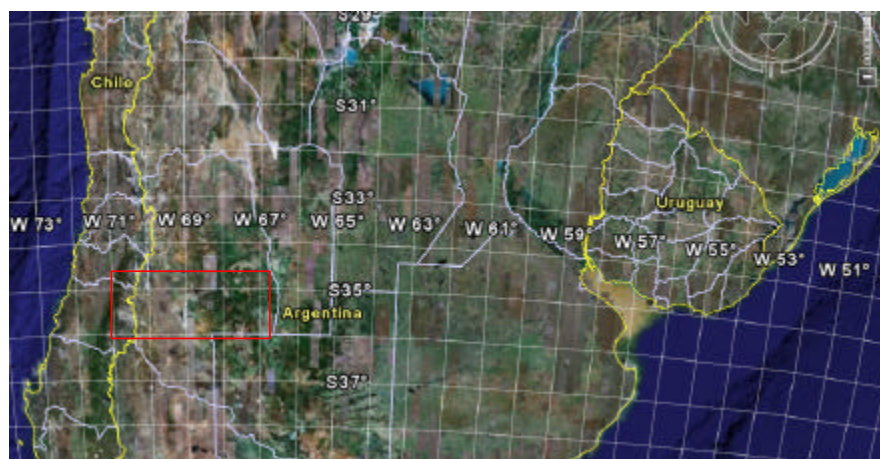


Figura 5 – Demarcación de la región geográfica de interés.

A pesar de los importantes cambios de pendiente en toda su extensión, se han logrado excelentes visualizaciones con simplificaciones con menos del 0,0004 % de los polígonos originales. La Figura 6 muestra seis aproximaciones de la región capturadas con una aplicación visualizadora que se desarrolló íntegramente en C++, cuya interfaz de usuario aprovecha las utilidades de la librería OpenGL [10]. La métrica de calidad de malla utilizada es la suma de las diferencias de cuadrados entre pares de cotas pertenecientes a la aproximación y la malla original cruda.

En principio, se unen los archivos HGT que definen la región de interés y luego comienza el proceso de simplificación poligonal del que resultan las aproximaciones con 10, 45, 845, 3549,

5873 y 5127 triángulos de la Figura 6. El indicador de error establece el nivel de detalle necesario en la región inicializando los tres factores que componen el criterio. A saber, el factor de tamaño del triángulo se inicializó como el 1% de la diagonal del rectángulo que representa al terreno original, el factor distancia corresponde a tomar el 80% del valor de distancia máxima y, finalmente, el factor asociado a la curvatura local queda establecido como el 20% del valor de curvatura máxima.

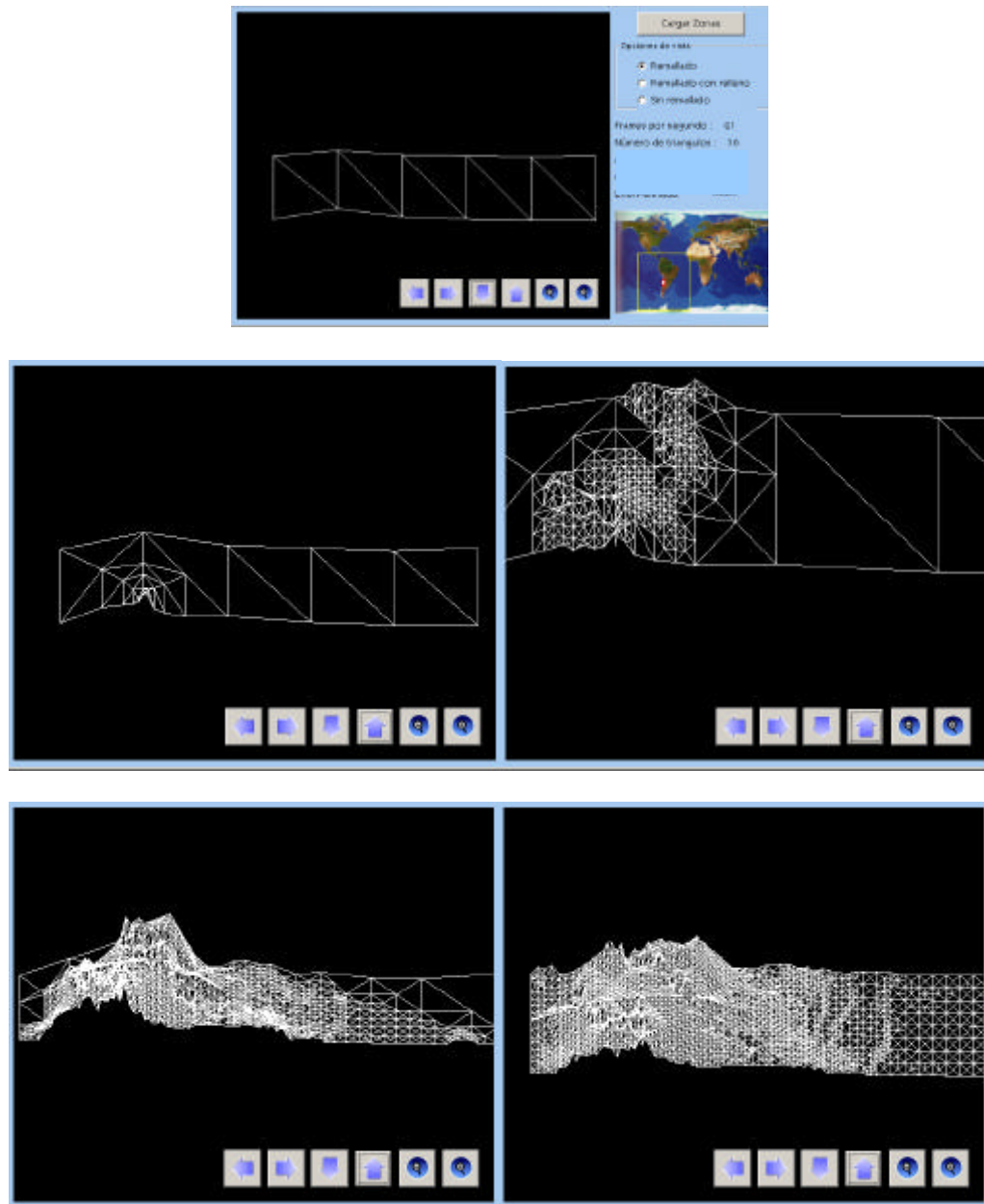


Figura 6 – Simplificación de topografía

5 CONCLUSIONES

Se presentó un algoritmo con una complejidad computacional de $O(n \log n)$ para la simplificación y visualización interactiva de modelos topográficos, con la cual se genera una jerarquía binaria de mallas poligonales del mismo terreno con distinta resolución. La selección automática de detalle se logra mediante un indicador que combina la posición del observador, el tamaño de los triángulos, sumado a un criterio de simplificación que es la curvatura local. Para evitar los cracks entre triángulos adyacentes, se agregan vértices ficticios que mantienen una cadena de dependencia con el vértice opuesto al triángulo analizado. Además se le suma una alternativa a la generación sistemática de triangle strip. El algoritmo de remallado mantiene la calidad visual tolerada por el usuario y detalla aún más las regiones dentro del campo de visión directo del observador. El algoritmo presentado simplifica notoriamente el volumen de datos a visualizar respecto de la grilla regular original.

REFERENCIAS

- [1] Abásolo, M.J., Blat, J., De Giusti, A., A Hierarchical Triangulation for Multiresolution Terrain Models. *Journal of Computer Science and Technology*, v.1, n.3, *Computers Science Research: State of the Art*, 2000.
- [2] Balmelli, L., Kovacevic, J., Vetterli, M., Quadrees for embedded surface visualization: constraints and efficient data structures. *In Proceedings of IEEE International Conference on Image Processing*, 2:487-491, 1999.
- [3] Cifuentes, M.V., D'Amato, J.P., García Bauza, C., Vénere, M.J., Clausse, A., Remallado interactivo como método para la navegación eficiente de modelos topográficos. *Mecánica Computacional*, 24:2385-2395, 2005.
- [4] Gross, M.H., Gatti, R., Staadt, O., Fast Multiresolution Surface, Meshing. *Proceedings of Visualization'95*, p. 135-142, 1995.
- [5] Hoppe, H., View-Dependent Refinement of Progressive Meshes. *Computers & Graphics. Proceedings of SIGGRAPH'97*, p.189-198, 1997.
- [6] Lindstrom, P., Pascucci, V., Visualization of Large Terrains Made Easy. *Proceedings of IEEE Visualization, San Diego, California*, p. 363-370, 2001.
- [7] Pajarola, R., Overview of quadtree-based terrain triangulation and visualization. *Technical Report, UCI-ICS-02-01, Information & Computer Science, University of California Irvine, ical Methods in Engineering*, 2002.
- [8] Rottger, S., Heidrich, W., Slusallek, P., and Seidel, H. P., Real-Time Generation of Continuous Levels of Detail for Height Fields. V. Skala, editor, *Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization*, pages 315-322, 1998.
- [9] Vénere, M.J., Cifuentes, M.V., D'Amato, J.P., García Bauza, C., Editor de escenarios para aplicaciones de Realidad Virtual. *34° JAIHO, Jornadas Argentinas de Informática e Investigación Operativa*, 2005.
- [10] Woo, M., Neider, J., Davis, T., OpenGL Programming Guide: The Official Guide to Learning OpenGL. *Addison-Wesley Developers Press*, 1997.
- [11] Xu Qing, Zhang Baoming, Tan Bing, Ma Dongyang, A Dynamic Multi-resolution Model and It's Application to Terrain Rendering. *Symposium on Geospatial Theory, Processing and Application. Ottawa*, 2002.